



Projet de Programmation Web

Création d'une application web

KADI Antonin
PIOT Mélanie
Classe FSI 1

Professeur : Louis CHEREL

Table des Matières

- 1) Description du Projet
- 2) Etape de Conception - Réalisation
- 3) Problèmes Rencontrées

Description du Projet

Le But du Projet est de réaliser une application web avec des technologies back-end et front-end en utilisant des frameworks pour différencier l'affichage du script.

Notre Projet est un site internet de comparaison d'Ordinateur. Dans le souhait de respecter au mieux le cahier des charges nous souhaitons multiplier les interactions avec l'utilisateur (Admin ou non). Ainsi vous pouvez, avant de comparer les ordinateurs, vous identifier, et choisir le type d'ordinateur que vous souhaitez. Une fois ces étapes faites vous pouvez si vous n'êtes qu'un simple utilisateur acheter un ordinateur et si vous êtes administrateur, ajouter, supprimer, ou modifier un ordinateur en plus.

Pour réaliser ce site internet, nous avons utilisé les langages HTML, CSS et Javascript (Vue.js) pour la partie front-end. Et Node.js (express) pour la partie back-end.

Voici un descriptif des étapes lorsque vous entrez sur le site internet :

Lorsque vous entrez dans le site internet vous devez vous identifier soit en tant qu'utilisateur, soit en tant qu'admin. Les choix d'interactions dépendent de votre authentification.

Lorsque vous êtes un utilisateur normal, vous pourrez choisir le type d'ordinateur que vous souhaitez voir. En fonction de votre choix, un Ordinateur différent vous sera proposer.

Puis vous arrivez avec un tableau qui affiche l'ordinateur correspondant à votre choix.

Vous pouvez ensuite décider d'acheter l'Ordinateur ou non.

Si vous êtes Administrateur, alors vous n'aurez pas le formulaire de choix, vous arriverez directement sur le tableau représentant tous les Ordinateurs.

Vous aurez alors le choix d'Ajouter un Ordinateur en remplissant toutes ses nouvelles caractéristiques. (1)

Vous pourrez aussi modifier un Ordinateur en remplissant la caractéristique à changer puis sa nouvelle valeur. (2)

Et enfin vous pourrez supprimer un Ordinateur du tableau. (3)

Nous avons aussi rajouté la possibilité de se reconnecter en cliquant sur le bouton à droite de la barre de navigation.

Etape de Conception-Réalisation

Pour réaliser cette application web, nous avons eu plusieurs étapes de conception et réalisation.

L'étape de conception a eu lieu au tout début du projet. Nous avons d'abord réfléchi au thème de l'application en réalisant quelques croquis.

Puis comme dit dans la description du projet, nous voulions au mieux respecter le cahier des charges. Nous avons donc décidé dans la mesure du réalisable d'avoir un maximum d'interaction avec l'utilisateur.

Une fois le design, le thème et le nombre d'interaction définis nous avons commencé la partie réalisation

Nous sommes passé par 3 étapes de réalisation.

La première étape, la plus courte, fut celle de la programmation en HTML et CSS. Elle a servi à poser les bases de notre site internet. Nous avons ainsi afficher séparément et statiquement toutes "nos pages" en mettant toutes les données en brutes. C'est surtout lors de cette phase que nous avons travaillé le design de l'application.

La deuxième étape s'est découpée en trois sous étapes:

La première fut celle d'intégrer du javascript afin de commencer les interactions et de rendre le site dynamique. Nous avons créer un enchaînement entre "nos pages" puis rendu dynamique nos formulaires d'ajout, de suppression et de modification.

La deuxième partie concerna le Vue.js. Une fois le javascript presque fini, nous avons dû le remplacer en intégrant petit à petit le Vue.js. Aujourd'hui il n'y a plus de code javascript dans le programme.

Enfin la dernière partie servit à incorporer le serveur avec node.js et express. Cette partie sert à garder en mémoire les modifications que l'on réalise mais aussi permet l'affichage de l'application (pour notre site l'affichage du tableau d'ordinateurs).

Problèmes Rencontrés

Le premier problème a concerné le framework Vue.js. Nous n'avions jamais utilisé Vue.js avant. Il nous a donc fallu appréhender le framework et comprendre son fonctionnement. Le Javascript et Vue.js ne fonctionnant pas de la même façon il a donc fallu s'adapter et changer notre façon de programmer et notre mode de réflexion.

De plus il ne fallait pas perdre trop de temps avec cette partie car la plus grosse, celle du back-end restait encore à être réalisé.

La deuxième difficultés concerna la partie serveur avec node.js et express fut la plus compliqué. Tout d'abord nous avons attendu que l'application fonctionne parfaitement en local avant d'intégrer le serveur.

Comme pour le Vue.js, il nous a fallu apprendre tout depuis le début.

Nous avons eu quelques problèmes pour afficher le tableau des Ordinateurs, mais ce problème a rapidement été résolu.

Le plus gros problème auquel nous avons fait face est la compréhension de node.js et de son fonctionnement. Il a fallu découvrir l'environnement node.js et express en très peu de temps ce qui ne nous a pas permis d'assimiler correctement son fonctionnement. Cela nous a fait perdre du temps lors de la conception du server.

Cette mauvaise compréhension de Node.js a entraîné une erreur lors de l'exécution de la fonction permettant d'ajouter un nouvel Ordinateur. Celle ci nous a fait perdre pratiquement 2 jours.

L'Erreur était une erreur 500 qui correspondait au server.

Cette erreur a été très complexe à résoudre car le serveur n'indique pas ce qui ne fonctionne pas mais seulement que quelque chose ne va pas. Après de nombreux essais nous avons compris que le problème venait des librairies node. Nous avons donc dû réinstaller node.js.

La fonction ajouter en elle-même n'a pas posé de soucis.

Pour la fonction Modifier au niveau du server, si nous mettions le paramètre à changer en brut dans la fonction (ex : listOrdi[i].nom), la modification s'effectuait correctement mais dès lors que nous mettions la valeur en variable (ex : listOrdi[i].req.params.type), la fonction ne marchait plus. Pris par le temps nous avons dû passer par `JSON.stringify` pour convertir la variable en donnée brute. Ne comprenant pas l'erreur, c'est la seule solution fonctionnelle que nous avons trouvée.